มุม dBASE

ใช้ dBASE III จัดเรียงคำภาษาไทย Thai Sort

ยืน ภู่วรวรรณ

ในการจัดเรียงคำนั้น ดีเบสมีคำสั่ง Sort และ Index อยู่แล้ว แต่การจัดเรียงนั้นจะใช้ได้สำหรับการจัดเรียงคำภาษาอังกฤษเท่านั้น เพราะคำในภาษาอังกฤษมีรูปแบบรหัส ASCII ที่จะสามารถนำมาเรียงได้โดยตรงโดยไม่มีปัญหาใด ๆ เกิดขึ้น แต่การเรียงคำในภาษาไทยนั้นมีหลักการที่แตกต่างออกไปมาก เพราะเรายึดหลักการจัดเรียงตามพจนานุกรมฉบับราชบัณฑิตยสถาน ปีพ.ศ. 2525ซึ่ง **ไม่ว่าจะจัดเรียงรหัสภาษาไทยอย่างไรก็ไม่สามารถใช้คำสั่ง Sort หรือ Index ในดีเบสมาใช้งานได้โดยตรง**

โครงสร้างอัลกอริทึมการเรียงคำไทยตามพจนานุกรม

หากเปิดพจนานุกรมฉบับราชบัณฑิตยสถานดู จะพบว่าในพจนานุกรมก็ไม่ได้บอกหลักเกณฑ์การเรียงไว้อย่างแน่ชัด หรือกำหนดเป็นสูตรที่แน่นอน แต่ด้วยหลักการที่ทำอยู่ ผู้เขียนสามารถนำมาประเมินเป็นกฎเกณฑ์เพื่อที่จะใช้งานได้ โดยลองดูการจัดเรียงในพจนานุกรมดู

การจัดเรียงคำนั้น จะพบว่า **ก่ำ** มาก่อน **กาง**, **ก่ำ ๆ** มาก่อน **กาง**, **เขน** มาก่อน **เข็น** คำว่า **เข็น** ก็มาก่อนคำว่า **เข่น** และเรามีการเรียงวรรณยุกต์เป็น **เกา เก่า เก้า** คำว่า **งัง ๆ** มาก่อน **งั่ง** คำว่า **แหง** มาก่อน **แหง่** และ **แหง่** มาก่อน **แห่ง** เป็นต้น

ด้วยกฎเกณฑ์นี้เอง เราจึงสร้างกลุ่มรหัสได้เป็น 3 กลุ่ม คือ

1. กลุ่มวรรณยุกต์ มีนัยสำคัญน้อยที่สุด จัดเรียงกันในกลุ่มเป็น ๆ ็ ่ ้ ๊ ๋ ์
2. กลุ่มสระมีนัยสำคัญรองลงมา ซึ่งจัดเรียงในกลุ่มเป็น ะ ั า ำ ิ ี ึ ื ุ ู เ แ โ ใ ไ
3. กลุ่มตัวอักขระมีนัยสำคัญสูงสุด เรียงตามลำดับก่อนหลังของพยัญชนะรวมตัว ฤ และ ฦ ด้วยดังนี้ ก ข ฃ ค ฅ ฆ ง จ ฉ ช ซ ฌ ญ ฎ ฏ ฐ ฑ ฒ ณ ด ต ถ ท ธ น บ ป ผ ฝ พ ฟ ภ ม ย ร ฤ ล ฦ ว ศ ษ ส ห ฬ อ ฮ

การจัดเรียงคำนี้ จะเปรียบเทียบในกลุ่มเรียงตามน้ำหนักมา เช่น คำว่า **กอ ก 'อ** จะเห็นว่า ก มีน้ำหนักเท่ากัน แต่ **กอ** ขึ้นก่อน **ก 'อ** หรือ **ก 'อ** ขึ้นก่อน **กอก**

ดังนั้นการให้น้ำหนักนี้ ถ้าต่างกลุ่มกันจะเทียบกันโดยตรงไม่ได้ เราจะต้องนำเอากลุ่มหนึ่งมาเทียบกันก่อน ถ้าเท่ากันจึงเทียบน้ำหนักในกลุ่ม 2 และเท่ากันอีกก็เทียบในกลุ่ม 3 ดังนั้นการจัดเรียงจึงต้องทำในลักษณะของการตรวจสอบเพื่อเปรียบเทียบ

อัลกอริทึมการเปรียบเทียบคำไทยนั้นผู้เขียนได้เสนอต่อสภาวิจัยแห่งชาติ เมื่อปี พ.ศ.2526 แล้ว (ดูเรื่องเทคนิคการพัฒนาภาษาไทย ตอน 4 ในวารสารฉบับนี้) สำหรับในที่นี้ผู้เขียนจะนำอัลกอริทึมนี้มาเขียนด้วยดีเบสเพื่อให้เห็นหลักการที่เด่นชัดขึ้น เพราะดีเบส III เขียนโปรแกรมรูปแบบ**โครงสร้าง**

1

```
B>type testthai.prg
*program test two thai string
*programmer yuen poovarawan
*date :1 may 1986
*
public sarastr,vanayuk,res
set procedure to thaisort
more = "y"
sarastr = "ะัาำิีึืุูเแโใไ"
vanayuk = "ๆ็่้๊๋์"
do while upper(more) = "Y"
   clear
   @ 10,1
   res = 0
   accept "Enter String A  :" to AST
   accept "Enter String B  :" to bst
   do thaicomp with ast,bst,res
   @ 15,0
   ?  "string A =",ast,"  string B =",bst,"result = ",res
   @ 20,0
   accept "more...." to more
enddo
close proc
return
```

โปรแกรมเปรียบเทียบสตริงไทย 2 ตัว

โปรแกรมดีเบส testthai ดังรูปที่ 1 นี้ เริ่มจากการกำหนด public ด้วยตัวแปร sarastr vanayuk และ res โดย sarastr เป็นอะเรย์ของสระ ซึ่งได้แก่ "ะ ั า ำ ิ ี ึ ื ุ ู เ แ โ ใ ไ" ส่วนอะเรย์ของ vanayuk ได้แก่ "ๆ ็ ่ ้ ๊ ๋ ์ ๎ ๏ ๚ ๛" สำหรับตัวแปร res คือตัวแปรที่จะบอกว่า เท่ากัน น้อยกว่า หรือมากกว่า โดยการแทนค่าเป็น 0, 1, 2 ตามลำดับ

ในที่นี้เราเริ่มจากการกำหนด res = 0 และรับข้อมูลภาษาไทยเข้ามาใน AST และ BST จากนั้นจึงส่งไปเปรียบเทียบใน thaicomp

การใช้ do thaicomp with ast, bst, res นั้นคือ ให้ไปกระทำในกระบวนความ thaicomp โดยผ่านค่าตัวแปร ast, bst และ res ไป โดยเมื่อเปรียบเทียบเสร็จแล้วจะได้ค่า res กลับมา ถ้า res = 0 คือเท่ากัน res = 1 คือ AST < BST ถ้า res = 2 คือ AST > BST การกระทำในกระบวนความนี้เรามีคำสั่ง Set procedure to thaisort ซึ่งการกระทำแบบกระบวนความนี้จะทำให้ดีเบสทำงานได้เร็วขึ้นกว่าเดิม

พิจารณาใน thaisort.Prg ซึ่งเป็น procedure file ดังรูปที่ 2

ใน procedure file นี้ มีกระบวนความอยู่ 2 กระบวนความคือ กระบวนความ Thaicomp และกระบวนความ sara กระบวนความ Thaicomp มีการกำหนดตัวพารามิเตอร์ไว้ 3 ตัว คือ A1, B1 และ THC สังเกตว่าตัวแปรในกระบวนความของดีเบสจะเป็นแบบท้องถิ่นเสมอ จะเห็นว่าการเปรียบเทียบใน Thaicomp นี้เหมือนกับอัลกอริทึมที่ผู้เขียนได้นำเสนอในรูปแบบคำสั่งลำลองแล้ว (ดูจากเรื่องพัฒนาการคอมพิวเตอร์ภาษาไทย

2

```
B>type thaisort.prg
PROCEDURE Thaicomp
PARAMETER A1,B1,THC
a = a1+chr(160)
b = b1+chr(160)
i=1
wa1 = 0
wa2 = 0
j=1
wb1 = 0
wb2 = 0
n1 = len(a)
n2 = len(b)
ok = .t.
DO WHILE i<=n1 .AND. j<=n2 .and. ok
   wa = substr(a,i,1)
   wb = substr(b,j,1)
   cwa = asc(wa)
   cwb = asc(wb)
   IF wa = " " .AND. wb = " "
      i = i + 1
      j = j + 1
   ELSE
      IF wa = " "
         thc = 1
         ok = .f.
      ELSE
          IF wb = " "
             thc = 2
             ok = .f.
          ELSE
             DO CASE
                CASE (cwa >=160 .and. cwa <=203) .and. ;
                     (cwb >= 160 .and. cwb <= 203)
                     IF cwa <> cwb
                        IF cwa < cwb
                           thc = 1
                        else
                           thc = 2
                        ENDIF
                        ok = .f.
                     ELSE
                        IF wa2 <> wb2
                           IF wa2 < wb2
                              thc = 1
                           ELSE
                              thc = 2
                           ENDIF
                           ok = .f.
                        ELSE
                           IF wa1 <> wb1
                              IF wa1< wb1
                                 thc = 1
                              ELSE
                                 thc = 2
                              ENDIF
                              ok = .f.
                           ELSE
                              i = i + 1
                              j = j + 1
                           ENDIF
                        ENDIF
                     ENDIF
                CASE (cwa >= 160 .and. cwa <=203) .and.  ;
                     (cwb < 160 .or. cwb > 203)
```

```
                    do sara with wb,wb1,wb2
                    j = j+ 1
               CASE (cwa < 160 .or. cwa > 203 ) .and. ;
                    (cwb >= 160 .and. cwb <= 203)
                    do sara with wa,wa1,wa2
                    i = i + 1
               OTHERWISE
                    do sara with wa,wa1,wa2
                    do sara with wb,wb1,wb2
                    i = i + 1
                    j = j + 1
            ENDCASE
         ENDIF
      ENDIF
   ENDIF
ENDDO
RETURN
*
PROCEDURE sara
PARAMETER w,w1,w2
k = 1
do while k <= 15
   if w = substr(sarastr,k,1)
      w2 = w2+k
      k  = 20
   endif
   k = k + 1
enddo
k = 1
do while k <=7
   if w = substr(vanayuk,k,1)
      w1 = w1 + k
      k = 10
   endif
   k = k+1
enddo
return
*
```

ตอน 4) ส่วนกระบวนความ sara เป็นการกำหนดหาค่าในอะเรย์ ซึ่งต้องลูปเข้าไปในอะเรย์ (ทำให้ช้า)

การเปรียบเทียบของ Thaistring 2 ตัวนี้ จะใช้เวลาประมาณ 10–40 วินาที ต่อการเปรียบเทียบครั้งหนึ่ง (เพราะดีเบสเป็นอินเตอร์พรีตเตอร์)

ใช้ Shell Sort กับ Thaicompare

เพื่อให้เห็นการใช้งานโดยตรงของการจัดเรียงข้อมูล เราต้องนำเอากระบวนความ Thaicomp มาผสมกับการจัดเรียง ในที่นี้ขอใช้วิธีการจัดเรียงแบบ Shell ซึ่งได้ยกตัวอย่างแบบ Quick sort ไว้ด้วยภาษาเบสิกแล้ว การจัดเรียงนี้จะสร้างไฟล์ Test.dbf โดยมีฟิลด์ภาษาไทยคือฟิลด์ key เมื่อป้อนข้อมูลภาษาไทยลงในฟิลด์นี้แล้ว เราจะเอามาจัดเรียง ก่อนการจัดเรียงเราจะ Copy to temp แล้วเอาไฟล์ temp มาจัดเรียง ผลลัพธ์อยู่ที่ไฟล์ temp

การจัดเรียงนี้ จะทำโดยการสลับเรคอร์ด หลังจากเปรียบเทียบกันแล้วด้วยคำสั่ง replace key with a1 (ดูในโปรแกรมรูปที่ 3) ตัวโปรแกรมนี้จะบันทึกเวลาเมื่อเริ่มต้นการจัดเรียงและเมื่อได้ทำการเรียงข้อมูลเสร็จแล้ว

สืบเนื่องจากการจัดเรียงเป็นการจัดเรียงบนแฟ้มข้อมูล temp โดยตรง จึงอาจจะต้องเสียเวลาบ้าง

ผลการจัดเรียงข้อมูลจะได้ดังนี้

```
data unsort

Record#   KEY
      1   กาง
      2   กา
      3   แหง
      4   แหง
      5   กวาง
      6   กวง
      7   แหง
      8   เจา
      9   เจา
     10   เจาว
start time : 00:40:06

finish time : 00:49:03

Record#   KEY
      1   กา
      2   กาง
      3   กวาง
      4   กวง
      5   เจา
      6   เจา
      7   เจาว
      8   แหง
      9   แหง
     10   แหง
```

9 นาที เรียงได้ 10 คำ

การเรียงในที่นี้ ใช้เวลาถึง 9 นาที เพื่อเรียงข้อมูล 10 คำ สาเหตุที่ช้าเพราะการทำงานทำบนฟลอบปี้ดิสก์ โดยเฉพาะการเปรียบเทียบกับอะเรย์ของสระและวรรณยุกต์ ซึ่งเราเปรียบเทียบหมดด้วยการวนรอบ ดีเบสจะทำงานได้ช้าเพราะเป็นอินเตอร์พรีตเตอร์

เหลือ 25 วินาที เมื่อใช้ clipper

แต่เมื่อเอาโปรแกรมนี้ไปทำการคอมพายล์ด้วย clipper ซึ่งเป็นดีเบส III คอมพายเลอร์ ผลทำให้เกิดโปรแกรม TSORT.EXE มีขนาดถึง 109 กิโลไบท์ และทำงานได้เร็วขึ้นมาก แต่ก็ใช้เวลาประมาณ 25 วินาที เมื่อจัดเรียงข้อมูล 10 ตัว

3

```
B>type tshell.prg
*program thaisort using saeree&yuen algorithm
*date :1 june 1986
*
public sarastr,vanayuk,thc,a1,b1
set talk off
set proc to thaisort
clear
use test
copy to  temp
use temp
? "data unsort"
?
list
? "start time :",time()
thc = 0
sarastr = "ะาำิีึืุูเแโใไ็่้๊๋์ั"
vanayuk = "กขฃคฅฆงจฉชซฌญฎฏฐฑฒณดตถทธนบปผฝพฟภมยรลวศษสหฬอฮ"
go bottom
n = recno()
d = recno()
do while d>1
   d = int(d/2)
   cntr = 0
   done = .f.
   do while .not. done
      cntr = cntr + 1
      done = .t.
      i1 = 1
      do while i1 <= (n-d)
         goto i1
         a1 = key
         goto i1 + d
         b1 = key
         thc = 0
         do thaicomp with a1,b1,thc
         if thc > 1
            repl key with a1
            goto i1
            repl key with b1
            done = .f.
         endif
         i1 = i1 + 1
      enddo
   enddo
enddo
?
? "finish time :",time()
?
list
return
```

ถ้าจะเรียงข้อมูลไทย+อังกฤษ

ในการเรียงข้อมูลภาษาไทยปนอังกฤษนั้น สามารถเปลี่ยนแปลงได้ในการเปรียบเทียบหาพยัญชนะ เช่นในกรณี cwa > = 160 และ cwa < = 203 ให้แก้ cwa > = 32 และแก้ a = a1+chr(160) เป็น a = a1+chr(32) และ b = b1+chr(32) แทนจะเห็นว่าการจัดเรียงรหัสถ้าให้ ก เริ่มที่ A1 หรือค่า 161 จะทำให้การจัดเรียงผสมไทยอังกฤษทำได้ทันที และถ้าพิจารณาจากโปรแกรมนี้ จะเป็นการจัดเรียงตามรหัสที่ทางห้องปฏิบัติการวิจัยไมโครคอมพิวเตอร์ใช้คือ 161-203 คือช่วงของพยัญชนะ และถ้าพิจารณาจากรหัสของสมอ. จะพบว่ารหัสสมอ. เหมาะสมกับการจัดเรียงคำไทย+อังกฤษเพราะกลุ่มตัวอักษรของสมอ. มีกลุ่มของอักษรอังกฤษ+พยัญชนะไทยอยู่ติดกัน คือเริ่มจากตัวภาษาอังกฤษไปจนถึง ก ข .....ฮ จากนั้น จะแบ่งตามกลุ่มของสระและวรรณยุกต์ตามลำดับ อย่างไรก็ตามอาจมีผู้คิดอัลกอริทึมการจัดเรียงแบบอื่นได้ แต่ก็จะต้องแยกตัวอักษรไทยออกเป็น 3 กลุ่มตามที่ผู้เขียนกล่าวอ้างไว้แล้วอยู่ดี

บทสรุปการใช้ดีเบสเรียงคำไทย

ผู้เขียนได้เขียนขึ้นมาเพื่อให้เห็นหลักการเรียงคำไทยที่ถูกต้องตามพจนานุกรม แต่ถ้าใช้ดีเบสจะช้ามาก ซึ่งมีวิธีการเพิ่มความเร็วขึ้นได้อีก แต่อย่างไรก็ตามหากใช้ดีเบสก็นับว่าช้าอยู่ดี ดังนั้นหากผู้ต้องการจัดเรียงคำไทย ผู้เขียนขอแนะนำให้ใช้คอมพายเลอร์ เช่น คลิปเปอร์ หรือใช้ภาษาอื่น เช่น เบสิก หรือปาสคาลเข้าช่วยจะทำได้เร็วขึ้น

มุมดีเบสนี้ จึงเน้นให้เห็นการประยุกต์ใช้งาน และหลักการทางด้านโปรแกรม ผู้อ่านสนใจจะลองดูก็ได้